Passage Retrieval with Vector Space and Query-Level Aspect Models

Raymond Wan[1], Vo Ngoc Anh[2], and Hiroshi Mamitsuka[1]

[1]Bioinformatics Center, Institute for Chemical Research, Kyoto University, Gokasho, Uji, 611-0011, Japan ({rwan,mami}@kuicr.kyoto-u.ac.jp)
[2]Department of Computer Science and Software Engineering, University of Melbourne, Victoria, 3010, Australia (vo@csse.unimelb.edu.au)

Abstract

This report describes the joint work by Kyoto University and the University of Melbourne for the TREC Genomics Track in 2007. As with 2006, the task for this year was the retrieval of passages from a biomedical document collection. The overall framework of our system from 2006 remains unchanged and is comprised of two parts: a paragraph-level retrieval system and a passage extraction system. These two systems are based on the vector space model and a probabilistic word-based aspect model, respectively. This year, we have adopted numerous changes to our 2007 system which we believe corrected some problems.

1 Introduction

Kyoto University and the University of Melbourne participated together for the TREC Genomics Track in 2007. As with last year's task, this year's aim was to retrieve passages from a full-text HTML collection of biomedical journals (see Hersh et al. [2006] for further details). While the collection this year remains unchanged, new queries were employed as well as modifications to the scoring scheme for passages.

Last year, we introduced our system for passage retrieval made of a paragraph-level retrieval system and a passage extraction system [Wan et al., 2006]. Since our performance last year was unsatisfactory, we re-designed parts of the system using what we learned from 2006. In Section 2, we begin with a brief description of our method. Section 3 describes our 2007 system by comparing against our 2006 system. Section 4 covers our three officially submitted runs and an additional set of runs that was performed after our results were released. Finally, we summarize this report in Section 5.

2 Method

The main task is to select passages from a collection of biomedical articles which are relevant to a query. Both the 2006 and 2007 tracks used the same collection of full papers from HighWire Press. Each article is in HTML and segmented into paragraphs (as indicated by the HTML <p> tags). Passages are sections of contiguous characters that do not include any of these paragraph boundaries. Performance is measured in terms of aspect, document, and passage retrieval.

Our method consists of two parts working together: a paragraph-level retrieval system and a passage extraction system. Based on the vector space model (VSM), the paragraph-level retrieval system constructs

an index off-line. When given a query, a ranked list of relevant paragraphs are identified and passed to the passage extraction system. By using scores obtained from a probabilistic word-based aspect model derived from word-pair co-occurrences, the contiguous sequence of words (a passage) from each paragraph that is most relevant to the query is returned. The paragraph score and the passage score are aggregated for passage re-ranking as the final step.

In the remainder of this section, we describe the two parts of this method in general terms. Implementation details are deferred to the section after.

2.1 Notation

We begin by defining some notation. The requirements of the Genomics Track makes it necessary to view the document collection as having a hierarchical structure. The document collection D is composed of a set of documents d, which are each made up of paragraphs $r \in R$. We assume that within a paragraph r, there is at most one relevant passage p. The set of words in D is W and the set of words that form a query q is Q. All words in the query are assumed to also be in the collection (i.e., $Q \in W$).

A word is represented as i for a query and j for a paragraph, such that $q = \{i\}$ and $r = \{j\}$. The paragraph frequency or the number of paragraphs that contain j is f_j. The within paragraph frequency of word j in paragraph r is $f_{r,j}$. Note that our terminology differs slightly from the information retrieval field since we are operating primarily at the paragraph-level. That is, even though the collection is based on articles, we view it as a collection of paragraphs.

The word-based aspect model makes use of the co-occurrence of two words. We make a distinction between the *co-occurrence* $n(i, j)$ and the *co-occurrence scores* $c(i, j)$. The co-occurrence $n(i, j)$ is the observed frequency of appearances of both i and j in the same paragraph throughout D. These values are used as input into our aspect model. The co-occurrence scores are the outputs from this process. In both matrices, $n(i, j)$ and $c(i, j)$ are undefined when $i = j$. The maximum co-occurrence score is denoted c_{max}.

2.2 Vector Space Model

The paragraph-level retrieval engine employed as the first component of our system has two functions: a) to generate the primary statistics needed for calculating the co-occurrence scores between pairs of terms, and b) to produce the first-round result list to each query, which will be further refined by the aspect model. The major function is the second one, where each original query is processed as a ranked query using the impact-based ranking approach.

The impact-based ranking approach is essentially a variation of the vector space model, where each distinct term of the collection is represented as a dimension in the n-dimension vector space (where n is the number of distinct terms in the collection). In this space, each document or query is represented by a vector, whose coordinates in a dimension is interpreted as the "importance" of the corresponding term in the document (or query). Traditionally, the coordinates are computed in a quantitative way and are floating-point values, and the level of similarity between a document and a query is defined as the cosine of the angle between the two respective vectors. In the impact-based approach, however, the coordinates (called impacts) are produced in a qualitative manner, and are integers values between 0 and (in this case) 8. Moreover, the level of similarity is now defined as the scalar product of the document and the query vector. Anh and Moffat [2005] describes the motivations and the details behind impact-based ranking.

2.3 Aspect Model

The aspect model (also, latent semantic analysis) has been proposed by others to associate words to documents [Deerwester et al., 1990, Hofmann et al., 1998]. As summarized by Deerwester et al. [1990], one-

mode factor analysis consists of a matrix of associations between all pairs of a single type of object. Deerwester et al. applies two-mode factor analysis where a matrix of two different types of objects is constructed. In their case, it is a term by document matrix which indicates the number of times term i occurs in document j. Using singular value decomposition, they reduce this matrix into k-dimensional space, and in doing so, show how latent semantic analysis (LSA) can be useful for document retrieval. Probabilistic latent semantic analysis (PLSA) [Hofmann, 2001] adds a probabilistic model to this earlier work by employing an iterative approach using the Expectation-Maximization (EM) algorithm [Dempster et al., 1977].

For our method, we return to one-mode factor analysis and combine this with PLSA. Instead of constructing a matrix of documents against documents, we build a matrix of words against words. We reduce this matrix to k clusters or latent states, where k is much less than the number of unique words in the collection (i.e., $k \ll |W|$).

The co-occurrence score $c(i, j)$ for the word pair (i, j) is obtained by summing across all of the latent states Z, which is of size $|Z| = k$:

$$c(i,j) = \sum_{z \in Z} p(i|z)p(j|z)p(z), \tag{1}$$

The parameters of this aspect model can be estimated using the EM algorithm by iterating between the following E-step and M-step:

E-step:

$$p(z|i,j) = \frac{p(i|z)p(j|z)p(z)}{\sum_{z' \in Z} p(i|z')p(j|z')p(z')} \tag{2}$$

M-step:

$$p(i|z) = \sum_{j \in W} n(i,j)p(z|i,j) \tag{3}$$

$$p(j|z) = \sum_{i \in Q} n(i,j)p(z|i,j) \tag{4}$$

$$p(z) = \sum_{i \in Q} \sum_{j \in W} n(i,j)p(z|i,j). \tag{5}$$

Initial values are generated at random using a uniform distribution. The output from the word-based aspect model (AM) is a set of scores $c(i, j)$ such that $c(i, j) = c(j, i)$, where $i \neq j$. Instead of assigning a score to $c(i, j)$ when $i = j$, we consider variations on the maximum co-occurrence score c_{max}.

3 TREC 2007 System Description

The organization of our system this year is shown in Figure 1. The inputs to the system are the document collection D and a query q. The output is a ranked list of passages. In the figure, "indexing" and "querying" refer to the paragraph-level retrieval system, while "passage extraction" and "score derivation" are parts of the passage extraction system. Re-ranking of passages by aggregating their paragraph and passage scores is performed as the final step, "score merging".

The parts of the paragraph-level retrieval system are self-explanatory. Score derivation refers to the use of the aspect model to assign scores to pairs of words. Passage extraction uses these scores to identify the words in the paragraph that are most relevant to the query. After the passages are extracted, as a requirement of the Genomics Track, their character positions in their respective documents are calculated using the Smith-Waterman sequence alignment algorithm [Smith and Waterman, 1981].

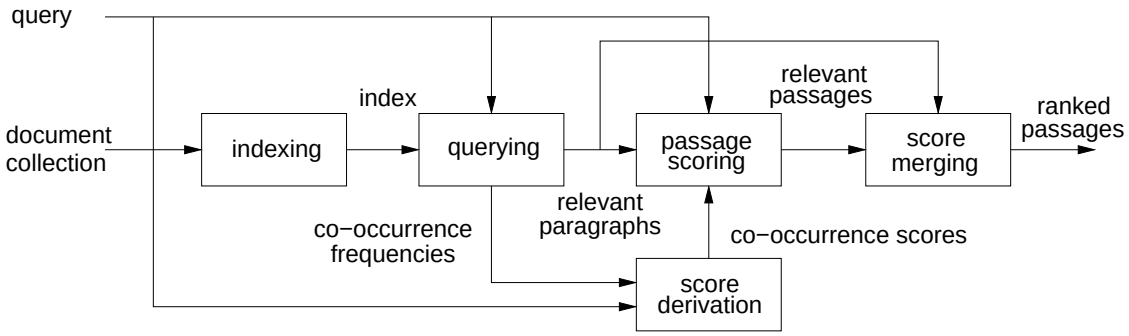


Figure 1: Our system includes a vector space model and an aspect model which include four components: indexing, querying, passage extraction, and score derivation. A final score merging steps performs passage re-ranking.

Feature	2006 System	2007 System
Aspect model	collection-level (off-line)	query-level (on-line)
Indexing level	article/paragraph	paragraph only
Look-up to external databases	yes	no
Stemming algorithm	Lovins	simplified stemming
Number of passages/paragraph	> 1	1

Table 1: Differences between our 2006 and 2007 systems.

3.1 System Differences

The main differences between our two systems are shown in Table 1 (see Wan et al. [2006] for a description of our last year's system.). After reporting our 2006 results, there were bugs in our system that were found and corrected for our 2007 system. We believe these problems affected the reliability of the co-occurrence scores produced by the aspect model.

The most significant difference this year is how the aspect model is used. Previously, the scores $c(i, j)$ were obtained off-line. Using last year's stemming rules, there were 1,299,308 unique words. At this size, the corresponding matrix and associated data structures for the aspect model would require significant amounts of memory. We address this issue by selecting a co-occurrence matrix of size n by n, where $n \ll |W|$. Last year, we chose all words that appeared in at least 1% of the collection of 162,259 articles ($n =$ 13,895).

While last year's method has the advantage of employing the aspect model off-line, it was found that much of the co-occurrence matrix was unused. Since the purpose of the matrix is to score a paragraph against a query in order to trim it to a more concise passage, applying this step on-line would be more sensible. Thus, the dimensions of the matrix chosen this year was m by n, where $m = |Q|$. Moreover, if one of the dimensions of the matrix is reduced, then the other dimension can be increased to $|W|$. This idea forms the basis of our implementation this year. The co-occurrence matrix is calculated on-line on a per-query basis instead of a per-collection basis. Unfortunately, this approach would make real-time querying infeasible.

Last year, we investigated both paragraph and document-level indexing in the retrieval system. Our results showed that document-level yielded poor effectiveness results. Because of this, we focused exclusively on paragraph-level retrieval.

We made extensive use of external databases in 2006 to expand the query through the addition of synonyms. We used the Biomedical abbreviation server, Entrez Gene, and Medical Subject Headings to expand

query terms for both querying and passage extraction (see Chang et al. [2002], Maglott et al. [2005], Nelson et al. [2004] for details about these sources of biological synonyms). While we no longer use these ideas this year, they remain possible options for our system in the future.

However, words are parsed in the same way as last year. A word is a contiguous sequence of alphabetic or numeric characters, but not a mix of both. So, "CD44" and "CD-44" are both divided into the two words "CD" and "44". Case-folding is also employed, as well as a stop-word list of 471 words. Instead of using the Lovins stemming algorithm [Lovins, 1968], we chose a simplified algorithm which considers only regular endings such as -s, -es, -ed, -ly, and -ing.

In addition to these differences, the formula used for scoring passages was completely changed, as we describe next.

3.2 Scoring Passages

We re-designed the scoring mechanism so that it more closely resembles work by others in information retrieval (see Witten et al. [1999] for details). The basic idea remains the same. Instead of calculating a single value (such as the cosine similarity) between the paragraph and the query, our aim is to assign a score to each paragraph word relative to the query words. If the two words *match* exactly, then some constant score is added to the paragraph. Otherwise, a smaller score is added for a *mismatch*. A passage is obtained from the paragraph by isolating the highest scoring section.

Last year's method for paragraph scoring involved breaking the paragraph into *sections*, where each section begins or ends with punctuation marks such as full stops and semi-colons. Then, *every* possible contiguous group of one or more sections is tested against the query. At the TREC 2006 conference, it was noticed that most groups extracted only one passage per paragraph. As our approach yielded little benefit and was too time consuming, we restrict each paragraph to produce only one passage this year. Punctuation marks are still used, but their use is deferred until later when character positions are calculated.

The score for a word j in paragraph r against the query q is defined as:

$$s(q, r, j) = \frac{|R|}{f_j} \times (1 + \log f_{r,j}) \sum_{i \in q} \bar{c}(i, j). \tag{6}$$

where $|R|$ is the number of paragraphs in the collection and $\bar{c}(i, j)$ is the score between two words i and j. We investigated two methods for calculating this score: Methods 1 and 2. Each method consists of two cases based on whether or not $i = j$.

$$\text{Method 1} \qquad \bar{c}(i, j) = \begin{cases} c_{\max} & \text{if } i = j \\ c(i, j) & \text{otherwise} \end{cases} \tag{7}$$

$$\text{Method 2} \qquad \bar{c}(i, j) = \begin{cases} \ln(1 + \frac{|R|}{f_i}) \times c_{\max} & \text{if } i = j \\ \ln(1 + \frac{|R|}{f_i}) \times c(i, j) & \text{otherwise} \end{cases} \tag{8}$$

The requirement of both methods is to keep the values between the two cases relatively close. If the constant score for a match is much larger than a mismatch, then the usefulness of the co-occurrence scores is minimized and the scoring scheme resembles a simple count of terms that appear in both the paragraph and the query. Because of this, we keep the match score to be the maximum value in the entire co-occurrence table ($c_{\max}$). Method 1 is our baseline where $c_{\max}$ and $c(i, j)$ are used without modification as the match and mismatch scores, respectively. Method 2 applies a scaling factor based on the inverse document frequency (IDF) of the query term f_i.

If each paragraph word's score was used immediately for isolating a passage, then each passage would be the longest contiguous sequence of words whose endpoints are words also found in the query (since they would have the highest scores). To rectify this, we update the score of each word j using its neighboring scores. We simply added half the score of each word's two immediate neighboring words and one quarter of the score of the two words which are two words away j, as depicted by Equation (9). In the future, we hope to investigate scoring mechanisms which employ a multiplication factor which decreases with increasing distance from j.

$$s'(q,r,j) = s(q,r,j) + \frac{1}{2}\left(s(q,r,j-1) + s(q,r,j+1)\right) + \frac{1}{4}\left(s(q,r,j-2) + s(q,r,j+2)\right) \quad (9)$$

With these updated scores, we obtain a passage by locating the two highest scoring words in the paragraph to act as endpoints of the passage. Note that "highest scoring words" can be any word and not necessarily words that are also query terms. The score of this passage p against the query q is divided by its length in words, according to Equation (10):

$$s(q,p) = \frac{1}{|p|} \times \sum_{j \in p} s'(q,j)\ . \quad (10)$$

Since punctuation marks are more natural boundaries for English texts, the final passage returned extends both endpoints to the nearest punctuation mark. However, the score of the passage excludes these extra characters. The last step is to map each passage to character positions using the Smith-Waterman algorithm.

4 Results

Hersh et al. [2006] reports that the collection consists of 12,641,127 legal spans (sequences of characters which did not contain any HTML paragraph tags). However, when indexed by our paragraph-level retrieval system using the case folding, stop-word list, and stemming rules described above, the number of paragraphs was $|R| = 10{,}234{,}783$. Thus, many documents were "empty" according to our system and were excluded. The number of unique words was $W = 1{,}480{,}399$. Both of these values are larger than last year due in part to our simplified stemming algorithm.

For all of the experiments described in this section, the parameters for the aspect model remained unchanged. The number of clusters was $k = 100$ clusters. The stopping condition of the EM algorithm was either a maximum number of iterations of 50 or when the difference in maximum likelihood between two consecutive iterations differed less than 0.0001. The Smith-Waterman algorithm was used using 1, -1, and 0 for the scores for a match, mismatch, and a gap, respectively.

Three submissions were evaluated as part of our participation in the Genomics Track. Also, we performed an additional set of runs which examined the effect from several variables.

4.1 Submitted Runs

The parameters for our official runs are shown in Table 2. We assign a name to each run in the first column. While the Genomics Track requires 1,000 results per query, our paragraph retrieval system can pass more than 1,000 paragraphs to the passage extraction system. The number of results provided is shown in the second column. Two scoring mechanisms were described in the previous section, and both were applied, as shown in the third column.

Both parts of our system assign a unique score. The paragraph-level retrieval system assigns a score to the entire paragraph. The passage retrieval system, though, assigns a score to the returned passage. These

ID	Number of results from VSM	Scoring method	VSM:AM	
kyoto1	1000	2	100	0
kyoto2	5000	2	0	100
kyoto3	5000	1	50	50

Table 2: The three official runs submitted by our group.

ID	Document	Aspect	Passage	Passage2
kyoto1	0.1892	0.1208	0.0474	0.0209
kyoto2	0.1191	0.0302	0.0235	0.0054
kyoto3	0.1022	0.0312	0.0204	0.0065
Minimum	0.0329	0.0197	0.0029	0.0008
Median	0.1897	0.1311	0.0565	0.0377
Mean	0.1862	0.1326	0.0560	0.0398
Maximum	0.3286	0.2631	0.0976	0.1148

Table 3: The results for our three official runs and additional statistics from all 66 official runs submitted to the TREC Genomics track for evaluation.

two scores are normalized and weighted such that the total weight is always 100%. A run with a weight of 100% is using only one of the systems to rank the final results.

Table 3 shows the results from our official runs as well as some statistics covering all 66 runs that were submitted for pooled evaluation. Retrieval effectiveness was measured in terms of document, aspect, and passage retrieval mean average precision (MAP) across all 36 queries using at most 1000 passages per query. Two methods of calculating passage retrieval were used. "Passage" is identical to last year's scheme, while "Passage2" is the official passage retrieval measure for this year.

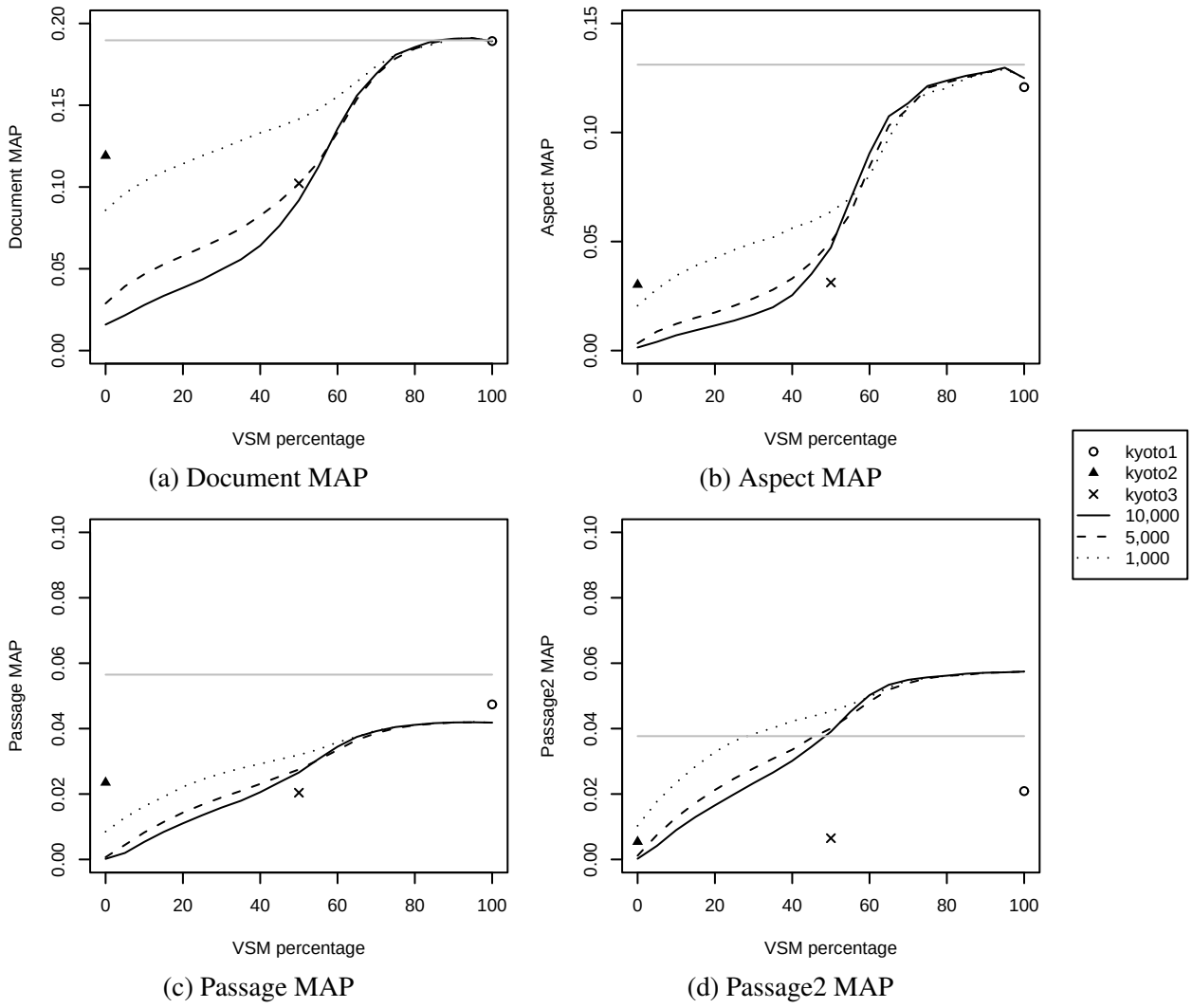
Out of our three runs, kyoto1 performed the best, but only average compared with runs by other groups. Our document retrieval performance was close to the median, while our other three scores were well below. The performance of the other two runs were significantly worse. From these results, we hypothesized that the cause of the problem with these two runs were the number of results from VSM, the scoring method, or the weight for score merging. We examined these three parameters with an additional set of runs which we call Varied-Method-2.

Also, after the submission of our official runs, we realized that there were problems in our implementation of the passage extraction system. There were errors in the scoring mechanism used by the passage extraction system and the alignment of passages to character positions using the Smith-Waterman algorithm. These problems were fixed and the changes are reflected in Varied-Method-2.

4.2 Additional Runs

A additional set of runs called Varied-Method-2 varied three different parameters. The results for the four measures are shown in Figure 2 with MAP plotted against the VSM percentage (0% means that AM is used exclusively to rank passages). Each of the three lines in each graph indicate the number of results supplied by the paragraph-level retrieval system. The horizontal gray line indicates the median MAP as reported in Table 3.

Our results for our official runs are also plotted on the graphs even though they are obtained from a system that had errors in it and thus, may not be comparable. Unfortunately, it would appear that fixing the problems actually degraded our system's performance. The only noticeable improvement was our Passage2



(a) Document MAP (b) Aspect MAP

(c) Passage MAP (d) Passage2 MAP

Figure 2: Mean average precision versus VSM percentage for the three sets of additional runs.

results. Combining the results of Table 3 and the trend of all of the lines, `kyoto3` should perform better than `kyoto2`. Since it does not, it implies that Method 1 performs worse and using the inverse document frequency seems required.

Excluding the Passage measure, all of our measures approach or slightly pass the median. The results show that an increase in the VSM percentage yields improved MAP in all cases. The optimal percentage appears to be around 95% or 100%. That is, the scores from the aspect model adversely affect the final ranking and are worse than the scores provided by the impact-based vector space model. Also, having less results from the VSM is better since the dotted line always performs the best in all four graphs. When more results are available to the passage extraction system, its scoring scheme moves irrelevant paragraphs into the top 1,000 passages for scoring, reducing our effectiveness.

5 Summary

In this report, we have described our work for the TREC 2007 Genomics Track. The two most notable changes to our system compared to last year was a shift to a query-based, on-line aspect model and a change in the passage scoring mechanism. Overall, our performance is average compared to other participants with MAP scores close to the median. An additional set of runs with a corrected system show that, unfortunately, our effectiveness has worsened slightly.

It would appear that the weight attributed to the VSM or the AM is one of the most significant factors affecting our method. The ranking provided by the VSM seems more useful, indicating that more work is required on the scoring regime used by the AM.

In the future we wish to examine both parts of the system more closely. So far, we have not investigated the effect from varying the parameters of the paragraph-level retrieval system. Variations, such as stemming rules, could have a noticeable effect on passage retrieval effectiveness. As for the aspect model, we have not properly evaluated varying values for the many parameters, such as the the number of latent states. Other future considerations include reducing the running time and the use of external biological databases to improve effectiveness.

Acknowledgements: We thank Ichigaku Takigawa (Kyoto University) for helpful discussions about the aspect model. RW was supported by a postdoctoral fellowship from the Japan Society for the Promotion of Science (JSPS). VNA was supported by the Australian Research Council's Center for Perceptive and Intelligent Machines in Complex Environments.

References

V. N. Anh and A. Moffat. Simplified similarity scoring using term ranks. In *Proc. 28th ACM International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 226–233, 2005.

J. T. Chang, H. Schütze, and R. B. Altman. Creating an online dictionary of abbreviations from MEDLINE. *Journal of the American Medical Informatics Association*, 9(6):612–620, November-December 2002.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

W. Hersh, A. M. Cohen, P. Roberts, and H. K. Rekapalli. TREC 2006 Genomics track overview. In *Proc. 15th Text Retrieval Conference (TREC 2006)*, November 2006.

T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1–2):177–196, January–February 2001.

T. Hofmann, J. Puzicha, and M. I. Jordan. Learning from dyadic data. In *Proc. 11th Conference on the Advances in Neural Information Processing Systems*, pages 466–472, 1998.

J. B. Lovins. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11 (1–2):22–31, 1968.

D. Maglott, J. Ostell, K. D. Pruitt, and T. Tatusova. Entrez Gene: Gene-centered information at NCBI. *Nucleic Acids Research*, 33:D54–D58, January 2005.

J. S. Nelson, M. Schopen, A. G. Savage, J.-L. Schulman, and N. Arluk. The MeSH translation maintenance system: Structure, interface design, and implementation. In *Proc. 11th World Congress on Medical Informatics*, pages 67–69, 2004.

T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

R. Wan, V. N. Anh, I. Takigawa, and H. Mamitsuka. Combining vector-space and word-based aspect models for passage retrieval. In E. M. Voorhees and L. P. Buckland, editors, *Proc. 15th Text Retrieval Conference (TREC 2006)*, Special Publication 500-272, November 2006.

I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes*. Morgan Kaufmann, second edition, 1999.